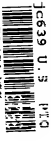
Docket No.: 43889-916

UTILITY PATENT APPLICATION
UNDER 37 CFR 1.53(b)



Box PATENT APPLICATION
Assistant Commissioner for Patents
Washington, DC 20231
Sir:

Transmitted herewith for filing is the patent application of:

INVENTOR: Masayuki YAMASAKI, Minoru OKAMOTO
FOR: SYSTEM AND METHOD FOR CONTROLLING PROGRAM EXECUTION

Enclosed are:

- ☒ 30 pages of specification, claims, abstract.
- ☒ Declaration and Power of Attorney.
- ☒ Priority Claimed.
- ☒ Certified copy of Japanese Patent Application No. 11-029568
- ☒ 12 sheets of formal drawing.
- ☒ An assignment of the invention to Matsushita Electric Industrial Co., Ltd. and the assignment recordation fee.
- ☒ An associate power of attorney.
- ☐ A verified statement to establish small entity status under 37 CFR 1.9 and 37 CFR 1.27.
- ☒ Information Disclosure Statement, Form PTO-1449 and reference.
- ☒ Return Receipt Postcard
- ☐

The filing fee has been calculated as shown below:

	NO. OF CLAIMS		EXTRA CLAIMS	RATE	AMOUNT
Total Claims	12	-20	0	$18.00	$0.00
Independent Claims	2	-3	0	$78.00	$0.00
				Multiple Dependent Claim(s)	$0.00
				Basic Fee	$690.00
				Total of Above Calculations	$690.00
				Less ½ for Small Entity	$0.00
				Assignment & Recording Fee	$40.00
				Total Fee	$730.00

☒ Please charge my Deposit Account No. 500417 in the amount of $730.00. A duplicate copy of this sheet is enclosed.

☒ The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 500417. A duplicate copy is enclosed.

☒ Any additional filing fees required under 37 CFR 1.16.

☒ The Commissioner is hereby authorized to charge payment of the following fees during the pendency of this application or credit any overpayment to Deposit Account No. 500417. A duplicate copy of this sheet is enclosed.

☒ Any patent application processing fees under 37 CFR 1.17.

☒ Any filing fees under 37 CFR 1.16 for presentation of extra claims.

Respectfully submitted,

MCDERMOTT, WILL & EMERY

Michael E. Fogarty
Registration No. 36,139

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 MEF:klm
Date: February 8, 2000
Facsimile: (202) 756-8087

SYSTEM AND METHOD FOR CONTROLLING PROGRAM EXECUTION

BACKGROUND OF THE INVENTION

The present invention relates to controlling the execution of a program in an information processor for processing instructions by pipelining.

In an information processor such as microprocessor or digital signal processor (DSP) for executing instructions by pipeline processing, a pipeline hazard resulting from conditional branching, i.e., a branch hazard, will happen. Particularly when there is a great number of pipeline stages, the branch hazard causes a serious problem.

To minimize the issuance of conditional branch instructions, conditionally executable instructions are adoptable effectively. For example, according to the technique disclosed in Japanese Laid-Open Publication No. 10-49368, each conditionally executable instruction contains a condition field specifying its own execution condition. That is to say, the instruction is selectively executed depending on whether or not the execution condition specified by its own condition field is satisfied. This technique is hard to apply to an information processor of the type using a short instruction word length, because each instruction to be executed by the processor cannot afford including such an additional condition field.

As for the short-word-length information processor, an instruction issued exclusively to control the conditional execution of succeeding instructions, i.e., an execution control instruction, is applicable effectively. For instance, according to the technique disclosed in Japanese Laid-Open Publication No. 7-253882, the execution control instruction contains a condition field specifying multiple registers. The number of registers specified in the condition field is always equal to the number of instructions, which succeed the execution control instruction and will be executed under controlled conditions, and is a fixed number. And it is determined based on the value of each of these registers whether or not a succeeding instruction corresponding to the register should be executed. As a result, conditional branch instructions do not have to be used so frequently.

According to the technique disclosed in the latter publication, however, if the number of instructions to be executed under controlled conditions should be increased, then the number of registers specified in the condition field of the execution control instruction has to be increased. As a result, the length of each instruction word must be increased unintentionally. Thus, the short-word-length information processor cannot control the conditional execution of so many succeeding instructions using the execution control instruction.

SUMMARY OF THE INVENTION

An object of the present invention is controlling the conditional execution of as many succeeding instructions as possible using an execution control instruction of a short word length to suppress the branch hazard in an information processor for processing the instructions by pipelining.

To achieve this object, the present invention introduces an instruction-specifying field, which is used to define, in binary code, the number of instructions that should be executed under controlled conditions, into the execution control instruction.

More specifically, the present invention uses an execution control instruction, which contains: a condition field specifying an execution condition; and an instruction-specifying field defining, in binary code, the number of instructions to be executed only conditionally. A decision is made as to whether or not the execution condition that has been specified by the condition field is satisfied. Based on the outcome of this decision, it is determined whether or not that number of instructions, which has been defined by the instruction-specifying field for instructions succeeding the execution control instruction, should be nullified.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram schematically illustrating

an exemplary configuration for a program execution control system according to the present invention.

Figure **2** illustrates a basic format for an execution control instruction according to the present invention.

Figure **3** illustrates a format for an execution control instruction according to a first embodiment of the present invention.

Figure **4** is a flowchart illustrating a program execution control procedure according to the first embodiment.

Figures **5A** and **5B** illustrate specific examples of the program execution control according to the first embodiment.

Figure **6** illustrates a format for an execution control instruction according to a second embodiment of the present invention.

Figure **7** is a flowchart illustrating a program execution control procedure according to the second embodiment.

Figures **8A** and **8B** illustrate specific examples of the program execution control according to the second embodiment.

Figure **9** illustrates a format for an execution control instruction according to a third embodiment of the present invention.

Figure **10** is a flowchart illustrating a program execution control procedure according to the third embodiment.

Figures **11A** and **11B** illustrate specific examples of the

program execution control according to the third embodiment.

Figure **12** illustrates a format for an execution control instruction according to a fourth embodiment of the present invention.

Figure **13** is a flowchart illustrating a program execution control procedure according to the fourth embodiment.

Figures **14A**, **14B**, **14C** and **14D** illustrate specific examples of the program execution control according to the fourth embodiment.

Figure **15** illustrates a format for an execution control instruction according to a fifth embodiment of the present invention.

Figure **16** is a flowchart illustrating a program execution control procedure according to the fifth embodiment.

Figures **17A**, **17B**, **17C** and **17D** illustrate specific examples of the program execution control according to the fifth embodiment.

DETAILED DESCRIPTION OF THE INVENTION

Figure **1** schematically illustrates an exemplary configuration for a program execution control system according to the present invention. The system shown in Figure **1** is used for controlling the execution of a program

in an information processor of the type processing instructions by pipelining. As shown in Figure 1, the system includes instruction providing section **10**, instruction executing section **20** and nullification controller **30**. The instruction providing section **10** includes program counter **11**, memory **12** and instruction register **13**. The program counter **11** sequentially specifies respective addresses of instructions to be fetched and executed. The memory **12** reads out the instructions, which together constitute a program including an execution control instruction, one after another in accordance with the addresses specified. And the instruction register **13** stores thereon the instructions read out one by one. An instruction set **INST** is provided from the instruction register **13** to the instruction executing section **20** and to the nullification controller **30**. The instruction executing section **20** includes: an arithmetic logic unit (ALU) **21** for performing arithmetic or logic operations in response to the instruction set **INST** provided; and a flag register **22** for holding multiple flags indicating the results of the computations. The nullification controller **30** receives control flags **CF** from the flag register **22**. In the illustrated embodiment, the control flags **CF** include two flags **F1** and **F2**, which have been set equal to zero or one by the ALU **21**.

Figure **2** illustrates a basic format for an execution control instruction according to the present invention. As

shown in Figure **2**, the execution control instruction contains instruction field **41**, condition field **42** and instruction-specifying field **43**. The instruction field **41** specifies the type of the instruction as an execution control instruction. The condition field **42** specifies an execution condition **EC**. And the instruction-specifying field **43** defines, in binary code, the number **N** (where N is a natural number) of instructions that will be executed only conditionally, i.e., just when the execution condition **EC** is satisfied. In response to the execution control instruction, the nullification controller **30** decides based on the control flags **CF** whether or not the execution condition **EC** specified by the condition field **42** is met. And based on the outcome of this decision, the nullification controller **30** determines whether or not the number **N** of instructions, which number has been defined by the instruction-specifying field **43** for instructions succeeding the execution control instruction, should be nullified. Suppose the nullification controller **30** has determined that the number **N** of instructions should be nullified since the execution condition **EC** is not met, the controller **30** asserts a nullification signal **NUL** to be supplied to the ALU **21**. In that case, even if the number **N** of instructions following the execution control instruction have already been executed to a midway point of the pipeline, these instructions are nullified in the ALU **21**. Accordingly,

the same results are attained as if NOP (no operation) instructions had been executed instead of these instructions. Alternatively, if the nullification controller **30** has determined that the number **N** of succeeding instructions should not be nullified since the execution condition **EC** is met, then the nullification signal **NUL** is not asserted. Thus, the number **N** of instructions succeeding the execution control instruction are enabled and executed by the ALU **21**.

Hereinafter, preferred embodiments of the present invention will be described with reference to Figures **3** through **17**.

EMBODIMENT 1

Figure **3** illustrates a 16-bit format of an execution control instruction according to a first embodiment of the present invention. As shown in Figure **3**, the execution control instruction contains a 12-bit instruction field **41**, a 2-bit condition field **42** and a 2-bit instruction-specifying field **43**. The instruction field **41** sets an operation code **OP** specifying the instruction type of the operation to be performed, i.e., execution control instruction. The condition field **42** specifies any of four types of execution conditions. In the illustrated embodiment, the execution condition **EC** to be specified is a binary code representing "F1 = 1", "F1 = 0", "F2 = 1" or "F2 = 0". The instruction-

specifying field **43** specifies, in binary code, the number **N** (where N=1 through 4) of instructions that will be executed under controlled conditions. In the illustrated embodiment, codes "00", "01", "10" and "11" represent "N=1", "N=2", "N=3" and "N=4", respectively.

Figure **4** illustrates a program execution control procedure according to the first embodiment. First, in Step **S1**, the execution condition **EC** and the number **N** of instructions to be executed under controlled conditions are defined for the condition field **42** and instruction-specifying field **43**, respectively. Next, in Step **S2**, the execution control instruction is executed. Then, in Step **S3**, it is determined based on the control flags **CF** whether or not the execution condition **EC** is satisfied. If the execution condition **EC** is not met, then the number **N** of instructions succeeding the execution control instruction are nullified in Step **S4**. Alternatively, if the execution condition **EC** is met, then the number **N** of instructions following the execution control instruction are executed in Step **S5**.

Figures **5A** and **5B** illustrate specific examples of the program execution control according to the first embodiment. In the illustrated examples, "F1 = 1" is the execution condition **EC** and "N = 4". As shown in Figure **5A**, if the execution condition is not met (i.e., when F1 = 0), then the four instructions succeeding the execution control

instruction (i.e., Succeeding Instructions 1 through 4) are nullified. On the other hand, if the execution condition is met (i.e., when F1=1), then the four instructions following the execution control instruction (i.e., Succeeding Instructions 1 through 4) are executed, not nullified, as shown in Figure **5B**.

In the first embodiment, the number of bits in the condition field **42** is two. Optionally, the number of execution conditions may be increased by increasing this bit number. The number of bits in the instruction-specifying field **43** is also two in the first embodiment. Alternatively, the number of instructions to be executed under controlled conditions may be increased by increasing this bit number. These modified examples will be equally applicable to all of the following second through fifth embodiments.

EMBODIMENT 2

Figure **6** illustrates a 16-bit format for an execution control instruction according to a second embodiment of the present invention. The format shown in Figure **6** is the same as that shown in Figure **3**.

Figure **7** illustrates a program execution control procedure according to the second embodiment. First, in Step **S11**, the execution condition **EC** and the number **N** of instructions to be executed under controlled conditions are

defined for the condition field **42** and instruction-specifying field **43**, respectively. Next, in Step **S12**, the execution control instruction is executed. Then, in Step **S13**, it is determined based on the control flags **CF** whether or not the execution condition **EC** is satisfied. If the execution condition **EC** is not met, then the number **N** of instructions succeeding the execution control instruction are nullified in Step **S14a**. Thereafter, the next number **N** of instructions following the former instruction set are executed in Step **S14b**. Alternatively, if the execution condition **EC** is met, then the number **N** of instructions succeeding the execution control instruction are executed in Step **S15a**. Thereafter, the next number **N** of instructions following the former instruction set are nullified in Step **S15b**.

Figures **8A** and **8B** illustrate specific examples of the program execution control according to the second embodiment. In the illustrated examples, "F1 = 1" is the execution condition **EC** and "N = 2". As shown in Figure **8A**, if the execution condition is not met (i.e., when F1 = 0), then the first two instructions succeeding the execution control instruction (i.e., Succeeding Instructions 1 and 2) are nullified, and the next two instructions (i.e., Succeeding Instructions 3 and 4) are executed. On the other hand, if the execution condition is met (i.e., when F1 = 1), then the first two instructions following the execution control

instruction (i.e., Succeeding Instructions 1 and 2) are executed, and the next two instructions (i.e., Succeeding Instructions 3 and 4) are nullified as shown in Figure **8B**. That is to say, according to this embodiment, instructions can be executed conditionally in accordance with the if-then-else construct.

EMBODIMENT 3

Figure **9** illustrates a 16-bit format for an execution control instruction according to a third embodiment of the present invention. As shown in Figure **9**, the execution control instruction contains a 10-bit instruction field **41**, a 2-bit condition field **42** and a 4-bit instruction-specifying field **43**. The instruction field **41** defines an operation code **OP** specifying the instruction type of the operation to be performed, i.e., execution control instruction. The condition field **42** specifies any of four types of execution conditions. The instruction-specifying field **43** consists of first and second instruction-specifying sub-fields **43a** and **43b**, each defining, in binary code, the number of instructions that will be executed only conditionally. The first instruction-specifying sub-field **43a** is a 2-bit field for defining, in binary code, the first number **N1** (where N1 = 1 through 4) of instructions to be executed under controlled conditions. The second instruction-specifying sub-field **43b** is also a 2-bit

field for defining, in binary code, the second number **N2** (where N2 = 1 through 4) of instructions to be executed under controlled conditions.

Figure **10** illustrates a program execution control procedure according to the third embodiment. First, in Step **S21**, the execution condition **EC** and the first and second numbers **N1** and **N2** of instructions to be executed under controlled conditions are defined for the condition field **42** and instruction-specifying sub-fields **43a** and **43b**, respectively. Next, in Step **S22**, the execution control instruction is executed. Then, in Step **S23**, it is determined based on the control flags **CF** whether or not the execution condition **EC** is satisfied. If the execution condition **EC** is not met, then the first number **N1** of instructions succeeding the execution control instruction are nullified in Step **S24a**. Thereafter, the second number **N2** of instructions following the first instruction set are executed in Step **S24b**. Alternatively, if the execution condition **EC** is met, then the first number **N1** of instructions succeeding the execution control instruction are executed in Step **S25a**. Thereafter, the second number **N2** of instructions following the first instruction set are nullified in Step **S25b**.

Figures **11A** and **11B** illustrate specific examples of the program execution control according to the third embodiment. In the illustrated examples, "F1 = 1" is the execution

condition **EC**, "N1 = 1" and "N2 = 3". As shown in Figure **11A**, if the execution condition is not met (i.e., when F1 = 0), then the first instruction succeeding the execution control instruction (i.e., Succeeding Instruction 1) is nullified, and the next three instructions (i.e., Succeeding Instructions 2, 3 and 4) are executed. On the other hand, if the execution condition is met (i.e., when F1 = 1), then the first instruction following the execution control instruction (i.e., Succeeding Instruction 1) is executed, and the next three instructions (i.e., Succeeding Instructions 2, 3 and 4) are nullified as shown in Figure **11B**. That is to say, according to this embodiment, instructions can be executed conditionally in accordance with the if-then-else construct. In addition, the respective numbers of instructions corresponding to the THEN and ELSE statements can be defined independently.

EMBODIMENT 4

Figure **12** illustrates a 16-bit format for an execution control instruction according to a fourth embodiment of the present invention. As shown in Figure **12**, the execution control instruction contains a 10-bit instruction field **41**, a 4-bit condition field **42** and a 2-bit instruction-specifying field **43**. The instruction field **41** defines an operation code **OP** specifying the instruction type of the operation to be

performed, i.e., execution control instruction. The condition field **42** consists of first and second condition sub-fields **42a** and **42b**, each specifying a single execution condition. The first condition sub-field **42a** is a 2-bit field for specifying any of four types of execution conditions as a first execution condition **EC1**. The second condition sub-field **42b** is also a 2-bit field for specifying any of four types of execution conditions as a second execution condition **EC2**. The instruction-specifying sub-field **43** defines, in binary code, the number **N** (where N = 1 through 4) of instructions to be executed under controlled conditions.

Figure **13** illustrates a program execution control procedure according to the fourth embodiment. First, in Step **S31**, the first and second execution conditions **EC1** and **EC2** and the number **N** of instructions to be executed under controlled conditions are defined for the first and second condition sub-fields **42a** and **42b** and instruction-specifying field **43**, respectively. Next, in Step **S32**, the execution control instruction is executed. Then, in Step **S33a**, it is determined based on the control flags **CF** whether or not the first execution condition **EC1** is satisfied. Next, in Step **S33b**, it is determined based on the control flags **CF** whether or not the second execution condition **EC2** is satisfied. If neither the first nor second execution condition **EC1** nor **EC2** is met, then the number **N** of instructions succeeding the execution control

instruction are nullified in Step **S34a**. Thereafter, the next number **N** of instructions succeeding the former instruction set are nullified in Step **S34b**. Alternatively, if the first execution condition **EC1** is not met and the second execution condition **EC2** is met, then the number **N** of instructions succeeding the execution control instruction are nullified in Step **S35a**. Thereafter, the next number **N** of instructions succeeding the former instruction set are executed in Step **S35b**. As another alternative, if the first execution condition **EC1** is met and the second execution condition **EC2** is not met, then the number **N** of instructions succeeding the execution control instruction are executed in Step **S36a**. Thereafter, the next number **N** of instructions succeeding the former instruction set are nullified in Step **S36b**. As further alternative, if both the first and second execution conditions **EC1** and **EC2** are met, then the number **N** of instructions succeeding the execution control instruction are executed in Step **S37a**. Thereafter, the next number **N** of instructions succeeding the former instruction set are executed in Step **S37b**.

Figures **14A** through **14D** illustrate specific examples of the program execution control according to the fourth embodiment. In the illustrated examples, "F1=1" is the first execution condition **EC1**, "F2 = 0" is the second execution condition **EC2** and "N=2". As shown in Figure **14A**, if neither

the first nor second execution condition is met (i.e., when F1 = 0 and F2 = 1), then four instructions succeeding the execution control instruction (i.e., Succeeding Instructions 1 through 4) are nullified. Alternatively, if the first execution condition is not met and the second execution condition is met (i.e., when F1 = 0 and F2 = 0), then two instructions succeeding the execution control instruction (i.e., Succeeding Instructions 1 and 2) are nullified and next two instructions succeeding the former instruction set (i.e., Succeeding Instructions 3 and 4) are executed as shown in Figure **14B**. As another alternative, if the first execution condition is met and the second execution condition is not met (i.e., when F1=1 and F2=1), then two instructions succeeding the execution control instruction (i.e., Succeeding Instructions 1 and 2) are executed and next two instructions succeeding the former instruction set (i.e., Succeeding Instructions 3 and 4) are nullified as shown in Figure **14C**. As further alternative, if both the first and second execution conditions are met (i.e., when F1 = 1 and F2 = 0), then four instructions succeeding the execution control instruction (i.e., Succeeding Instructions 1 through 4) are executed, not nullified, as shown in Figure **14D**.

It should be noted that the number **M** of condition sub-fields in the execution control instruction may be equal to or larger than three. In such a case, a number **N** of

instructions, which number has been defined by the instruction-specifying field **43** for instructions succeeding the execution control instruction, are regarded as instructions to be executed under controlled conditions. If an execution condition **ECm** (where m=1 through M), which has been specified by associated one of the number **M** of condition sub-fields, is not met, then the step of nullifying the number **N** of instructions at a location corresponding to the specified execution condition **ECm** is performed. Furthermore, if an execution condition **ECm** (where m=1 through M), which has been specified by associated one of the number **M** of condition sub-fields, is met, then the step of executing the number **N** of instructions at a location corresponding to the specified execution condition **ECm** is performed.

EMBODIMENT 5

Figure **15** illustrates a 16-bit format for an execution control instruction according to a fifth embodiment of the present invention. As shown in Figure **15**, the execution control instruction contains an 8-bit instruction field **41**, a 4-bit condition field **42** and a 4-bit instruction-specifying field **43**. The instruction field **41** defines an operation code **OP** specifying the instruction type of the operation to be performed, i.e., execution control instruction. The condition field **42** consists of first and second condition sub-fields

42a and **42b**, each specifying a single execution condition. The first condition sub-field **42a** is a 2-bit field for specifying any of four types of execution conditions as a first execution condition **EC1**. The second condition sub-field **42b** is also a 2-bit field for specifying any of four types of execution conditions as a second execution condition **EC2**. The instruction-specifying field **43** consists of first and second instruction-specifying sub-fields **43a** and **43b**, each defining, in binary code, the number of conditionally executable instructions. The first instruction-specifying sub-field **43a** is a 2-bit field for defining, in binary code, the first number **N1** (where N1 = 1 through 4) of instructions to be executed under controlled conditions. The second instruction-specifying sub-field **43b** is also a 2-bit field for defining, in binary code, the second number **N2** (where N2 = 1 through 4) of instructions to be executed under controlled conditions.

Figure **16** illustrates a program execution control procedure according to the fifth embodiment. First, in Step **S41**, the first and second execution conditions **EC1** and **EC2** and first and second numbers **N1** and **N2** of conditionally executable instructions are defined for the first and second condition sub-fields **42a** and **42b** and first and second instruction-specifying sub-fields **43a** and **43b**, respectively. Next, in Step **S42**, the execution control instruction is executed. Then, in

Step **S43a**, it is determined based on the control flags **CF** whether or not the first execution condition **EC1** is satisfied. Next, in Step **S43b**, it is determined based on the control flags **CF** whether or not the second execution condition **EC2** is satisfied. If neither the first nor second execution condition **EC1** nor **EC2** is met, then the number **N1** of instructions succeeding the execution control instruction are nullified in Step **S44a**. Thereafter, the next number **N2** of instructions succeeding the former instruction set are nullified in Step **S44b**. Alternatively, if the first execution condition **EC1** is not met and the second execution condition **EC2** is met, then the number **N1** of instructions succeeding the execution control instruction are nullified in Step **S45a**. Thereafter, the next number **N2** of instructions succeeding the former instruction set are executed in Step **S45b**. As another alternative, if the first execution condition **EC1** is met and the second execution condition **EC2** is not met, then the number **N1** of instructions succeeding the execution control instruction are executed in Step **S46a**. Thereafter, the next number **N2** of instructions succeeding the former instruction set are nullified in Step **S46b**. As further alternative, if both the first and second execution conditions **EC1** and **EC2** are met, then the number **N1** of instructions succeeding the execution control instruction are executed in Step **S47a**. Thereafter, the next number **N2** of instructions succeeding the

former instruction set are executed in Step **S47b**.

Figures **17A** through **17D** illustrate specific examples of the program execution control according to the fifth embodiment. In the illustrated examples, "F1 = 1" is the first execution condition **EC1**, "F2 = 0" is the second execution condition **EC2**, "N1 = 1" and "N2 = 3". As shown in Figure **17A**, if neither the first nor second execution condition is met (i.e., when F1 = 0 and F2 = 1), then four instructions succeeding the execution control instruction (i.e., Succeeding Instructions 1 through 4) are nullified. Alternatively, if the first execution condition is not met and the second execution condition is met (i.e., when F1=0 and F2 = 0), then one instruction succeeding the execution control instruction (i.e., Succeeding Instruction 1) is nullified and the next three instructions following the former instruction (i.e., Succeeding Instructions 2, 3 and 4) are executed as shown in Figure **17B**. As another alternative, if the first execution condition is met and the second execution condition is not met (i.e., when F1=1 and F2=1), then one instruction succeeding the execution control instruction (i.e., Succeeding Instruction 1) is executed and the next three instructions succeeding the former instruction (i.e., Succeeding Instructions 2, 3 and 4) are nullified as shown in Figure **17C**. As further alternative, if both the first and second execution conditions are met (i.e., when F1=1 and F2=

0), then four instructions succeeding the execution control instruction (i.e., Succeeding Instructions 1 through 4) are executed, not nullified, as shown in Figure **17D**.

It should be noted that the number **M** of condition sub-fields of the execution control instruction may be equal to or larger than three. In such a case, the same number **M** of instruction-specifying sub-fields, which correspond to the number **M** of condition sub-fields, respectively, are included in the instruction-specifying field of the execution control instruction. A number **Nm** of instructions (where m=1 through M), which number has been defined by associated one of the number **M** of instruction-specifying sub-fields for instructions succeeding the execution control instruction, are regarded as instructions to be executed under controlled conditions. If an execution condition **ECm** (where m = 1 through M), which is specified by associated one of the number **M** of condition sub-fields, is not met, then the step of nullifying the number **Nm** of instructions at a location corresponding to the specified execution condition **ECm** is performed. Furthermore, if the execution condition **ECm** (where m = 1 through M), which has been specified by associated one of the number **M** of condition sub-fields, is met, then the step of executing the number **Nm** of instructions at a location corresponding to the specified execution condition **ECm** is performed.

WHAT IS CLAIMED IS:

1. A system for controlling the execution of a program in an information processor for processing instructions by pipelining, the system comprising:

means for providing a set of instructions including an execution control instruction, the execution control instruction containing a condition field and an instruction-specifying field, the condition field specifying an execution condition, the instruction-specifying field defining, in binary code, the number of instructions to be executed only conditionally;

means for deciding whether or not the execution condition that has been specified by the condition field is satisfied; and

means for determining based on the outcome of the decision whether or not said number of instructions, which number has been defined by the instruction-specifying field for instructions succeeding the execution control instruction, should be nullified.

2. A method for controlling the execution of a program in an information processor for processing instructions by pipelining, the method comprising the steps of:

a) providing an execution control instruction, the execution control instruction containing a condition field

and an instruction-specifying field, the condition field specifying an execution condition, the instruction-specifying field defining, in binary code, the number of instructions to be executed only conditionally;

b) deciding whether or not the execution condition that has been specified by the condition field is satisfied; and

c) determining based on the outcome of the decision step b) whether or not said number of instructions, which number has been defined by the instruction-specifying field for instructions succeeding the execution control instruction, should be nullified.

3. The method of claim 2, wherein the condition field is a single field for specifying the execution condition, and

wherein the instruction-specifying field is a single field for defining the instruction number, and

wherein the step c) comprises the sub-step of regarding said number of instructions, which number has been defined by the instruction-specifying field for instructions succeeding the execution control instruction, as instructions to be executed only conditionally, and nullifying the conditionally executable instructions if the execution condition that has been specified by the condition field is not satisfied.

4. The method of claim 3, wherein the step c) further

comprises the sub-step of executing the conditionally executable instructions if the execution condition that has been specified by the condition field is satisfied.

5. The method of claim 2, wherein the condition field is a single field for specifying the execution condition, and

wherein the instruction-specifying field is a single field for defining the instruction number, and

wherein the step c) comprises the sub-steps of:

regarding said number of instructions, which number has been defined by the instruction-specifying field for instructions succeeding the execution control instruction, as a first set of conditionally executable instructions, and nullifying the first set of conditionally executable instructions if the execution condition that has been specified by the condition field is not satisfied; and

regarding said number of instructions, which number has been defined by the instruction-specifying field for instructions succeeding the first set of conditionally executable instructions, as a second set of conditionally executable instructions, and nullifying the second set of conditionally executable instructions if the execution condition that has been specified by the condition field is satisfied.

6. The method of claim 5, wherein the step c) further comprises the sub-steps of:

getting the first set of conditionally executable instructions executed if the execution condition that has been specified by the condition field is satisfied; and

getting the second set of conditionally executable instructions executed if the execution condition that has been specified by the condition field is not satisfied.

7. The method of claim 2, wherein the condition field is a single field for specifying the execution condition, and

wherein the instruction-specifying field contains first and second instruction-specifying sub-fields, which respectively define, in binary code, first and second numbers of instructions to be executed only conditionally, and

wherein the step c) comprises the sub-steps of:

regarding the first number of instructions, which number has been defined by the first instruction-specifying sub-field for instructions succeeding the execution control instruction, as a first set of conditionally executable instructions, and nullifying the first set of conditionally executable instructions if the execution condition specified by the condition field is not satisfied; and

regarding the second number of instructions, which number has been defined by the second instruction-specifying

sub-field for instructions succeeding the first set of conditionally executable instructions, as a second set of conditionally executable instructions, and nullifying the second set of conditionally executable instructions if the execution condition specified by the condition field is satisfied.

8. The method of claim 7, wherein the step c) further comprises the sub-steps of:

getting the first set of conditionally executable instructions executed if the execution condition that has been specified by the execution field is satisfied; and

getting the second set of conditionally executable instructions executed if the execution condition that has been specified by the execution field is not satisfied.

9. The method of claim 2, wherein the condition field includes multiple condition sub-fields, each specifying a single associated execution condition, and

wherein the instruction-specifying field is a single field for defining the instruction number, and

wherein the step c) comprises a plurality of sub-steps, and

wherein in each said sub-step, said number of instructions, which number has been defined by the

instruction-specifying field for instructions succeeding the execution control instruction, are regarded as conditionally executable instructions, and, if the execution condition specified by associated one of the condition sub-fields is not satisfied, the conditionally executable instructions at a location corresponding to the execution condition specified are nullified.

10. The method of claim 9, wherein the step c) comprises a plurality of sub-steps,

wherein in each said sub-step, if the execution condition specified by associated one of the condition sub-fields is satisfied, the conditionally executable instructions at a location corresponding to the execution condition specified are executed.

11. The method of claim 2, wherein the condition field includes multiple condition sub-fields, each specifying a single execution condition, and

wherein the instruction-specifying field includes multiple instruction-specifying sub-fields corresponding to the respective condition sub-fields, each said instruction-specifying sub-field defining an associated instruction number in binary code, and

wherein the step c) comprises a plurality of sub-steps,

and

wherein in each said sub-step, said number of instructions, which number has been defined by associated one of the instruction-specifying sub-fields for instructions succeeding the execution control instruction, are regarded as conditionally executable instructions, and, if the execution condition specified by associated one of the condition sub-fields is not satisfied, the conditionally executable instructions at a location corresponding to the execution condition specified are nullified.

12. The method of claim 11, wherein the step c) comprises a plurality of sub-steps,

wherein in each said sub-step, if the execution condition specified by associated one of the condition sub-fields is satisfied, the conditionally executable instructions at the location corresponding to the execution condition specified are executed.

ABSTRACT OF THE DISCLOSURE

An execution control instruction is applied to an information processor of the type processing instructions by pipelining to suppress the occurrence of branch hazard. The execution control instruction contains: a condition field for specifying an execution condition; and an instruction-specifying field for defining, in binary code, the number of instructions to be executed only conditionally. In response to the execution control instruction, a nullification controller decides, based on control flags provided from an arithmetic logic unit, whether or not the execution condition specified by the condition field is satisfied. And based on the outcome of this decision, the controller determines whether or not that number of instructions, which has been defined by the instruction-specifying field for instructions succeeding the execution control instruction, should be nullified. If the controller has determined that the specified number of succeeding instructions should be nullified since the execution condition is not met, then the controller asserts a nullification signal to be supplied to the arithmetic logic unit. In this manner, a large number of succeeding instructions are executable conditionally using an execution control instruction of a short word length.

Fig. 1

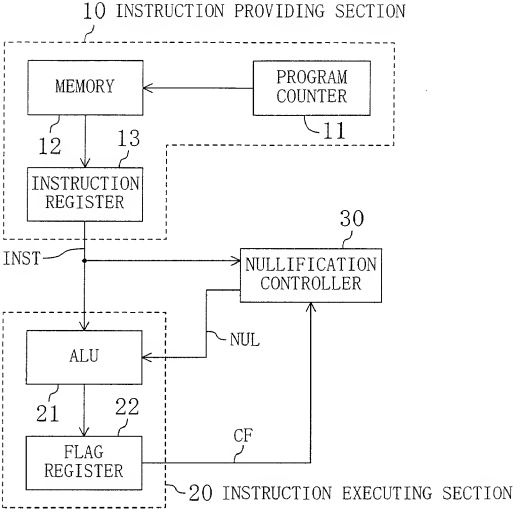

10 INSTRUCTION PROVIDING SECTION
MEMORY
PROGRAM COUNTER
11
12
13
INSTRUCTION REGISTER
30
INST
NULLIFICATION CONTROLLER
NUL
ALU
21
22
CF
FLAG REGISTER
20 INSTRUCTION EXECUTING SECTION

Fig. 2

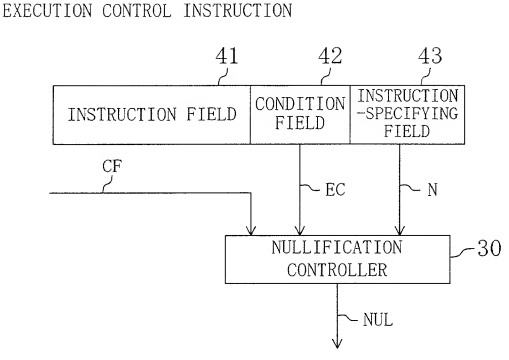

EXECUTION CONTROL INSTRUCTION
41
42
43
INSTRUCTION FIELD
CONDITION FIELD
INSTRUCTION -SPECIFYING FIELD
CF
EC
N
NULLIFICATION CONTROLLER
30
NUL

Fig. 3

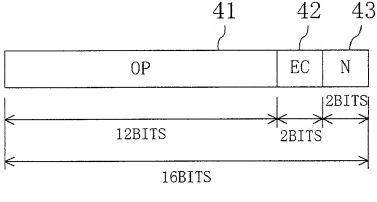

41
42
43
OP
EC
N
2BITS
12BITS
2BITS
16BITS

Fig. 4

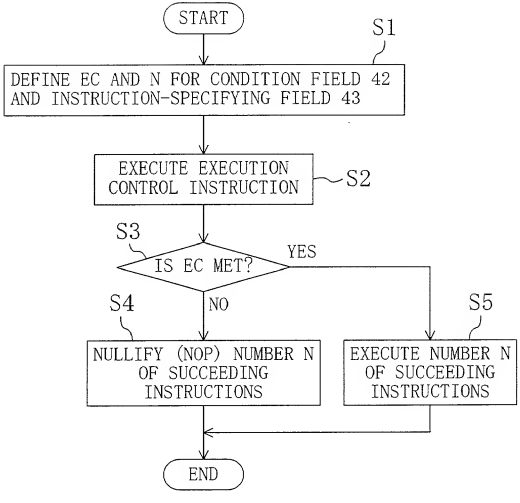

START
S1
DEFINE EC AND N FOR CONDITION FIELD 42
AND INSTRUCTION-SPECIFYING FIELD 43
EXECUTE EXECUTION
CONTROL INSTRUCTION
S2
S3
IS EC MET?
YES
NO
S4
S5
NULLIFY (NOP) NUMBER N
OF SUCCEEDING
INSTRUCTIONS
EXECUTE NUMBER N
OF SUCCEEDING
INSTRUCTIONS
END

Fig. 15

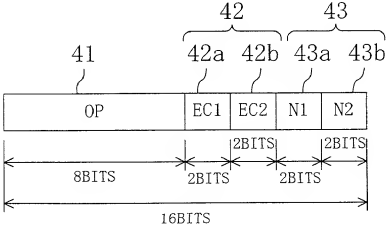

42
43
41
42a
42b
43a
43b
OP
EC1
EC2
N1
N2
2BITS
2BITS
8BITS
2BITS
2BITS
16BITS

Fig. 5A

```
IF EC IS NOT MET (F1=0)
    EXECUTION CONTROL INSTRUCTION F1, 4
    NOP
    NOP
    NOP
    NOP
    SUCCEEDING INSTRUCTION 5
```

Fig. 5B

```
IF EC IS MET (F1=1)
    EXECUTION CONTROL INSTRUCTION F1, 4
    SUCCEEDING INSTRUCTION 1
    SUCCEEDING INSTRUCTION 2
    SUCCEEDING INSTRUCTION 3
    SUCCEEDING INSTRUCTION 4
    SUCCEEDING INSTRUCTION 5
```

Fig. 6

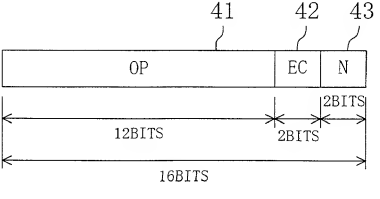


Fig. 7

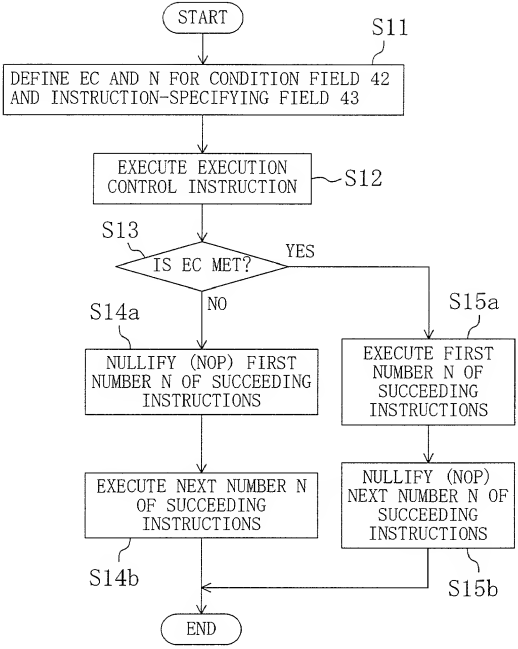

START
S11
DEFINE EC AND N FOR CONDITION FIELD 42 AND INSTRUCTION-SPECIFYING FIELD 43
EXECUTE EXECUTION CONTROL INSTRUCTION
S12
S13
IS EC MET?
YES
NO
S14a
NULLIFY (NOP) FIRST NUMBER N OF SUCCEEDING INSTRUCTIONS
EXECUTE NEXT NUMBER N OF SUCCEEDING INSTRUCTIONS
S14b
S15a
EXECUTE FIRST NUMBER N OF SUCCEEDING INSTRUCTIONS
NULLIFY (NOP) NEXT NUMBER N OF SUCCEEDING INSTRUCTIONS
S15b
END

Fig. 8A

```
IF EC IS NOT MET (F1=0)
    EXECUTION CONTROL INSTRUCTION F1,2
    NOP
    NOP
    SUCCEEDING INSTRUCTION 3
    SUCCEEDING INSTRUCTION 4
    SUCCEEDING INSTRUCTION 5
```

Fig. 8B

```
IF EC IS MET (F1=1)
    EXECUTION CONTROL INSTRUCTION F1,2
    SUCCEEDING INSTRUCTION 1
    SUCCEEDING INSTRUCTION 2
    NOP
    NOP
    SUCCEEDING INSTRUCTION 5
```

Fig. 9

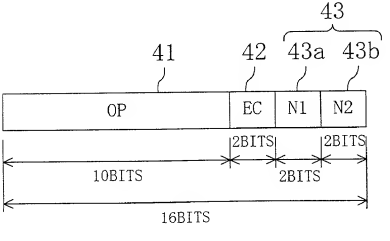


Fig. 10

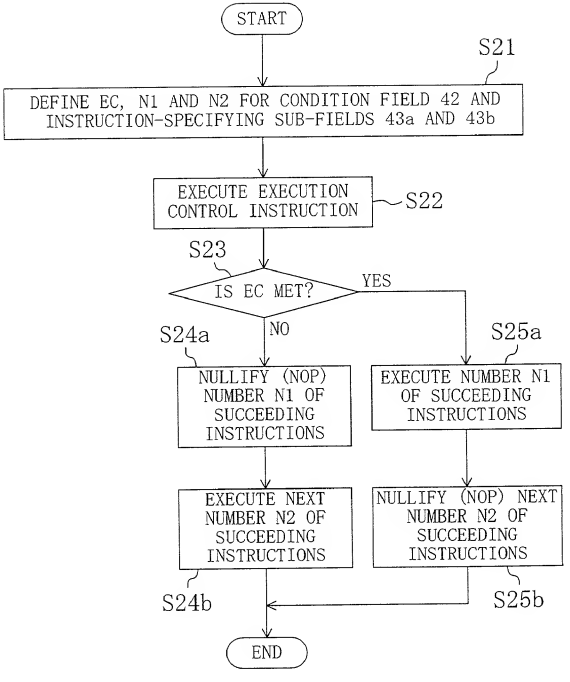

START
S21
DEFINE EC, N1 AND N2 FOR CONDITION FIELD 42 AND INSTRUCTION-SPECIFYING SUB-FIELDS 43a AND 43b
EXECUTE EXECUTION CONTROL INSTRUCTION
S22
S23
IS EC MET?
YES
NO
S24a
S25a
NULLIFY (NOP) NUMBER N1 OF SUCCEEDING INSTRUCTIONS
EXECUTE NUMBER N1 OF SUCCEEDING INSTRUCTIONS
EXECUTE NEXT NUMBER N2 OF SUCCEEDING INSTRUCTIONS
NULLIFY (NOP) NEXT NUMBER N2 OF SUCCEEDING INSTRUCTIONS
S24b
S25b
END

Fig. 11A

```
IF EC IS NOT MET (F1=0)
    EXECUTION CONTROL INSTRUCTION F1, 1, 3
    NOP
    SUCCEEDING INSTRUCTION 2
    SUCCEEDING INSTRUCTION 3
    SUCCEEDING INSTRUCTION 4
    SUCCEEDING INSTRUCTION 5
```

Fig. 11B

```
IF EC IS MET (F1=1)
    EXECUTION CONTROL INSTRUCTION F1, 1, 3
    SUCCEEDING INSTRUCTION 1
    NOP
    NOP
    NOP
    SUCCEEDING INSTRUCTION 5
```

Fig. 12

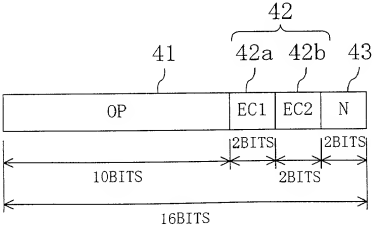


Fig. 13

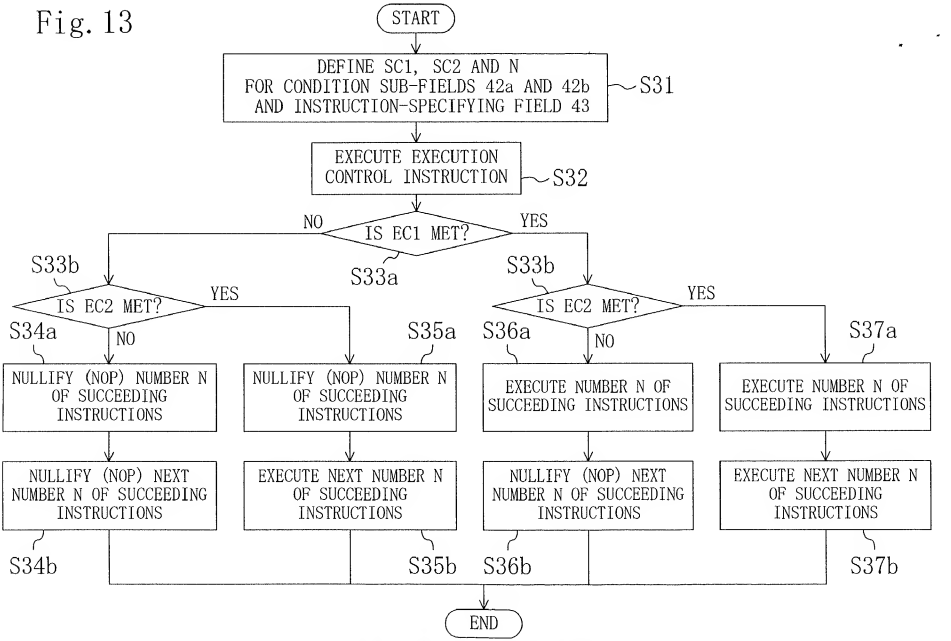

START
DEFINE SC1, SC2 AND N FOR CONDITION SUB-FIELDS 42a AND 42b AND INSTRUCTION-SPECIFYING FIELD 43
S31
EXECUTE EXECUTION CONTROL INSTRUCTION
S32
IS EC1 MET?
NO
YES
S33a
S33b
IS EC2 MET?
YES
NO
S34a
NULLIFY (NOP) NUMBER N OF SUCCEEDING INSTRUCTIONS
NULLIFY (NOP) NEXT NUMBER N OF SUCCEEDING INSTRUCTIONS
S34b
S35a
NULLIFY (NOP) NUMBER N OF SUCCEEDING INSTRUCTIONS
EXECUTE NEXT NUMBER N OF SUCCEEDING INSTRUCTIONS
S35b
S33b
IS EC2 MET?
YES
NO
S36a
EXECUTE NUMBER N OF SUCCEEDING INSTRUCTIONS
NULLIFY (NOP) NEXT NUMBER N OF SUCCEEDING INSTRUCTIONS
S36b
S37a
EXECUTE NUMBER N OF SUCCEEDING INSTRUCTIONS
EXECUTE NEXT NUMBER N OF SUCCEEDING INSTRUCTIONS
S37b
END

Fig. 14A

```
IF NEITHER EC1 NOR EC2 IS MET (WHEN F1=0 AND F2=1)
        EXECUTION CONTROL INSTRUCTION F1,NF2,2
        NOP
        NOP
        NOP
        NOP
        SUCCEEDING INSTRUCTION 5
```

Fig. 14B

```
IF EC1 IS NOT MET AND EC2 IS MET (WHEN F1=0 AND F2=0)
        EXECUTION CONTROL INSTRUCTION F1,NF2,2
        NOP
        NOP
        SUCCEEDING INSTRUCTION 3
        SUCCEEDING INSTRUCTION 4
        SUCCEEDING INSTRUCTION 5
```

Fig. 14C

```
IF EC1 IS MET AND EC2 IS NOT MET (WHEN F1=1 AND F2=1)
        EXECUTION CONTROL INSTRUCTION F1,NF2,2
        SUCCEEDING INSTRUCTION 1
        SUCCEEDING INSTRUCTION 2
        NOP
        NOP
        SUCCEEDING INSTRUCTION 5
```

Fig. 14D

```
IF BOTH EC1 AND EC2 ARE MET (WHEN F1=1 AND F2=0)
        EXECUTION CONTROL INSTRUCTION F1,NF2,2
        SUCCEEDING INSTRUCTION 1
        SUCCEEDING INSTRUCTION 2
        SUCCEEDING INSTRUCTION 3
        SUCCEEDING INSTRUCTION 4
        SUCCEEDING INSTRUCTION 5
```

Fig. 16

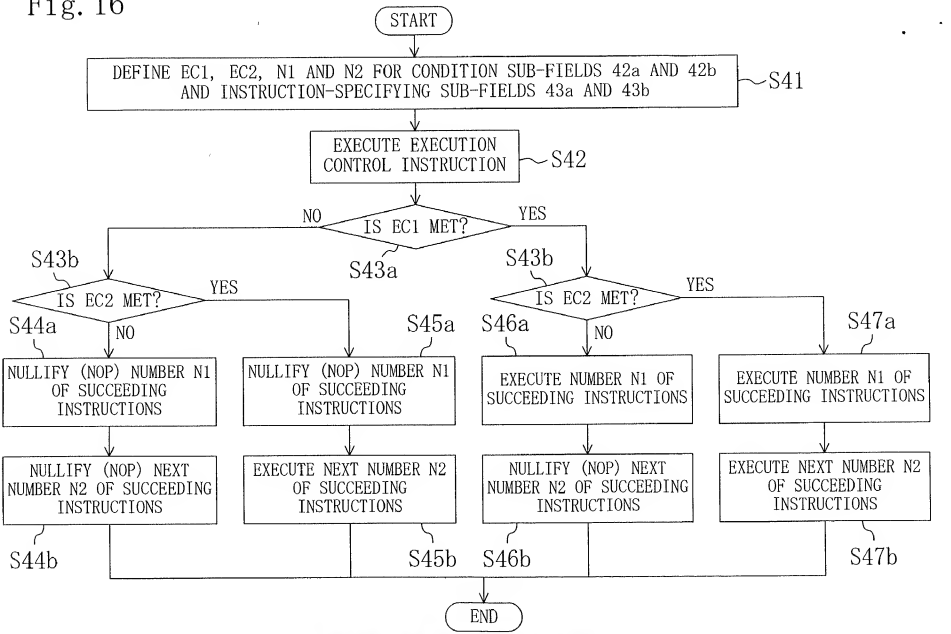

START
DEFINE EC1, EC2, N1 AND N2 FOR CONDITION SUB-FIELDS 42a AND 42b AND INSTRUCTION-SPECIFYING SUB-FIELDS 43a AND 43b
S41
EXECUTE EXECUTION CONTROL INSTRUCTION
S42
IS EC1 MET?
NO
YES
S43a
S43b
IS EC2 MET?
YES
NO
S44a
NULLIFY (NOP) NUMBER N1 OF SUCCEEDING INSTRUCTIONS
NULLIFY (NOP) NEXT NUMBER N2 OF SUCCEEDING INSTRUCTIONS
S44b
S45a
NULLIFY (NOP) NUMBER N1 OF SUCCEEDING INSTRUCTIONS
EXECUTE NEXT NUMBER N2 OF SUCCEEDING INSTRUCTIONS
S45b
S43b
IS EC2 MET?
YES
NO
S46a
EXECUTE NUMBER N1 OF SUCCEEDING INSTRUCTIONS
NULLIFY (NOP) NEXT NUMBER N2 OF SUCCEEDING INSTRUCTIONS
S46b
S47a
EXECUTE NUMBER N1 OF SUCCEEDING INSTRUCTIONS
EXECUTE NEXT NUMBER N2 OF SUCCEEDING INSTRUCTIONS
S47b
END

Fig. 17A

```
IF NEITHER EC1 NOR EC2 IS MET (WHEN F1=0 AND F2=1)
        EXECUTION CONTROL INSTRUCTION F1,NF2,1,3
        NOP
        NOP
        NOP
        NOP
        SUCCEEDING INSTRUCTION 5
```

Fig. 17B

```
IF EC1 IS NOT MET AND EC2 IS MET (WHEN F1=0 AND F2=0)
        EXECUTION CONTROL INSTRUCTION F1,NF2,1,3
        NOP
        SUCCEEDING INSTRUCTION 2
        SUCCEEDING INSTRUCTION 3
        SUCCEEDING INSTRUCTION 4
        SUCCEEDING INSTRUCTION 5
```

Fig. 17C

```
IF EC1 IS MET AND EC2 IS NOT MET (WHEN F1=1 AND F2=1)
        EXECUTION CONTROL INSTRUCTION F1,NF2,1,3
        SUCCEEDING INSTRUCTION 1
        NOP
        NOP
        NOP
        SUCCEEDING INSTRUCTION 5
```

Fig. 17D

```
IF BOTH EC1 AND EC2 ARE MET (WHEN F1=1 AND F2=0)
        EXECUTION CONTROL INSTRUCTION F1,NF2,1,3
        SUCCEEDING INSTRUCTION 1
        SUCCEEDING INSTRUCTION 2
        SUCCEEDING INSTRUCTION 3
        SUCCEEDING INSTRUCTION 4
        SUCCEEDING INSTRUCTION 5
```

Docket No. _____

COMBINED DECLARATION/POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled SYSTEM AND METHOD FOR CONTROLLING PROGRAM EXECUTION, the specification of which

(check one) X is attached hereto.

___ was filed on _____ as Application Serial No. _____.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, § 1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed
11-029568 (Number)	JAPAN (Country)	08/02/1999 (Day/Month/Year Filed)	X Yes ___ No
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	___ Yes ___ No
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	___ Yes ___ No

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

1999-P-1757/S1

(Appln. Serial No.)	(Filing Date)	(Status-patented, pending, abandoned)
(Appln. Serial No.)	(Filing Date)	(Status-patented, pending, abandoned)

I hereby appoint as my attorneys, with full power of substitution and revocation, to prosecute the patent application identified above and to transact all business in the U.S. Patent and Trademark Office connected therewith: Raphael V. Lupo (Reg. No. 28,363); Jack Q. Lever, Jr. (Reg. No. 28,149); Kenneth L. Cage (Reg. No. 26,151); Stanislaus Aksman (Reg. No. 28,562); Paul Devinsky (Reg. No. 28,553); Edward E. Kubasiewicz (Reg. No. 30,020), Michael E. Fogarty (Reg. No. 36,139); Brian E. Ferguson (Reg. No. 36,801); Robert W. Zelnick (Reg. No. 36,976); and Wilhlem F. Gadiano (Reg. No. 37,136).

Please address all correspondence and telephone calls to:

Jack Q. Lever, Jr.
McDERMOTT, WILL & EMERY
600 Thirteenth Street, N.W.
Washington, D.C. 20005-3096
(202) 756-8000

The undersigned hereby authorizes the U.S. attorneys named herein to accept and follow instructions from Maeda Patent Office as to any action to be taken in the Patent and Trademark Office regarding this application without direct communication between the U.S. attorney and the undersigned. In the event of a change in the persons from whom instructions may be taken, the U.S. attorneys named herein will be so notified by the undersigned.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor Masayuki YAMASAKI

Inventor's signature Masayuki Yamasaki Date Jan. 31, 2000
Residence* Osaka, Japan Citizenship Japan
Post Office Address 22-707, Shinmachi, Daito-shi, Osaka 574-0037, Japan

Full name of second inventor Minoru OKAMOTO

Inventor's signature Minoru Okamoto Date Jan. 31, 2000

Residence* Osaka, Japan Citizenship Japan

Post Office Address 2-21-20-503, Minamikaneda, Suita-shi, Osaka 564-0044, Japan

* City and State, or City and Country for foreign inventors

Docket No.: 43889-916 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	
Masayuki YAMASAKI, et al.	:	
Serial No.:	:	Group Art Unit:
Filed: February 08, 2000	:	Examiner:

For: SYSTEM AND METHOD FOR CONTROLLING PROGRAM EXECUTION

ASSOCIATE POWER OF ATTORNEY

Honorable Commissioner of
Patents and Trademarks
Washington, D. C. 20231

Sir:

The undersigned Principal Attorney of record hereby appoints the following Attorneys as his Associates with regard to the above-identified application: Edward A. Becker, Reg. No. 37,777; Stephen A. Becker, Reg. No. 26,527; Marcel K. Bingham, Reg. No. 42,327; John G. Bisbikis, Reg. No. 37,095; Carl L. Brandt, Reg No. 44,555, Daniel Bucca, Reg. No. 42,368; Kenneth L. Cage, Reg. No. 26,151; Carina M. Tan, Reg. No. P-45,769; Stephen C. Carlson, Reg. No. 39,929; Jennifer Chen, Reg. No. 42,404; Thomas A. Corrado, Reg. No. 42,439; Lawerence T. Cullen, Reg. No. 44,489; Paul Devinsky, Reg. No. 28,553; Margaret M. Duncan, Reg. No. 30,879; Brian E. Ferguson, Reg. No. 36,801; Michael E. Fogarty, Reg. No. 36,139; John R. Fuisz, Reg. No. 37,327; Willem F. Gadiano, Reg. No. 37,136; Keith E. George, Reg. No. 34,111; Matthew V. Grumbling, Reg. No. 44,427; John A. Hankins, Reg. No. 32,029; Brian D. Hickman, Reg. No. 35,894; Joseph Hyosuk Kim, Reg. No. 41,425; Eric J. Kraus, Reg. No. 36,190; Jack Q. Lever, Reg. No. 28,149; Raphael V. Lupo, Reg. No. 28,363; Christine F. Martin, Reg. No. 39,762; Michael A. Messina, Reg. No. 33,424; Eugene J. Molinelli, Reg. No. 42,901; Christopher J. Palermo, Reg. No. 42,056; Dawn L. Palmer, Reg. No. 41,238; Joseph H. Paquin, Jr., Reg. No. 31,647; William D. Pegg, Reg. No. 42,988; Robert L. Price, Reg. No. 22,685; Thomas D. Robbins, Reg. No. 43,669; Gene Z. Rubinson, Reg. No. 33,351; Joy Ann G. Serauskas, Reg. No. 27,952; David A. Spenard, Reg. No. 37,449; Arthur J. Steiner, Reg. No. 26,106; David L. Stewart, Reg. No. 37,578; Michael D. Switzer, Reg. No. 39,552; Leonid D. Thenor, Reg. No. 39,397; Keith J. Townsend, Reg. No. 40,358; Daniel S. Trainor, Reg. No. 43,959; Cameron K.

Weiffenbach, Reg. No. 44,488; Aaron Weisstuch, Reg. No. 41,557; Edward J. Wise, Reg. No. 34,523; Alexander V. Yampolsky, Reg. No. 36,324; and Robert W. Zelnick, Reg. No. 36,976 all of

McDERMOTT, WILL & EMERY
600 13th Street, N.W.
Washington, DC 20005-3096

Please continue to address all communications to the undersigned.

February 8, 2000

By_____
Michael E. Fogarty
Registration No. 36,139
Attorney for Applicant